

1. Record Nr.	UNINA9910451455903321
Autore	Powell Gavin
Titolo	Oracle performance tuning for 10gR2 [[electronic resource] /] / Gavin Powell
Pubbl/distr/stampa	Burlington, MA, : Digital Press, c2007
ISBN	1-281-00716-1 9786611007164 0-08-049202-9
Edizione	[2nd ed.]
Descrizione fisica	1 online resource (955 p.)
Altri autori (Persone)	PowellGavin
Disciplina	005.75/85
Soggetti	Relational databases Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Previous ed. published: New York : Digital Press, c2004, under title: Oracle high performance tuning for 9i and 10g. Includes index.
Nota di contenuto	Front Cover; Oracle ® Performance Tuning for 10gR2; Copyright Page; Contents; Preface; Introduction; Part I: Data Model Tuning; Chapter 1. The Relational Database Model; 1.1 The Formal Definition of Normalization; 1.2 A Layperson's Approach to Normalization; 1.3 Referential Integrity; Chapter 2. Tuning the Relational Database Model; 2.1 Normalization and Tuning; 2.2 Referential Integrity and Tuning; 2.3 Optimizing with Alternate Indexes; 2.4 Undoing Normalization; Chapter 3. Different Forms of the Relational Database Model; 3.1 The Purist's Relational Database Model 3.2 Object Applications and the Relational Database ModelChapter 4. A Brief History of Data Modeling; 4.1 The History of Data Modeling; 4.2 The History of Relational Databases; 4.3 The History of the Oracle Database; 4.4 The Roots of SQL; Part II: SQL Code Tuning; Chapter 5. What Is SQL?; 5.1 DML and DDL; 5.2 Transaction Control; 5.3 Parallel Queries; Chapter 6. The Basics of Efficient SQL; 6.1 The SELECT Statement; 6.2 Using Functions; 6.3 Pseudocolumns; 6.4 Comparison Conditions; Chapter 7. Advanced Concepts of Efficient SQL; 7.1 Joins; 7.2 Using Subqueries for Efficiency

7.3 Using Synonyms; 7.4 Using Views; 7.5 Temporary Tables; 7.6 Resorting to PL/SQL; 7.7 Object and Relational Conflicts; 7.8 Replacing DELETE with TRUNCATE; Chapter 8. Common-Sense Indexing; 8.1 What and How to Index; 8.2 Types of Indexes; 8.3 Types of Indexes in Oracle Database; 8.4 Tuning BTree Indexes; 8.5 Summarizing Indexes; Chapter 9. Oracle SQL Optimization and Statistics; 9.1 What Is the Parser?; 9.2 What Is the Purpose of the Optimizer?; 9.3 Rule-Based versus Cost-Based Optimization; Chapter 10. How Oracle SQL Optimization Works; 10.1 Data Access Methods; 10.2 Sorting; 10.3 Special Cases; Chapter 11. Overriding Optimizer Behavior Using Hints; 11.1 How to Use Hints; 11.2 Hints: Suggestion or Force?; 11.3 Classifying Hints; 11.4 Influencing the Optimizer in General; 11.5 Naming Query Blocks for Hints; Chapter 12. How to Find Problem Queries; 12.1 Tools to Detect Problems; 12.2 EXPLAIN PLAN; 12.3 SQL Trace and TKPROF; 12.4 TRCSSESS; 12.5 Autotrace; 12.6 Oracle Database Performance Views for Tuning SQL; Chapter 13. Automated SQL Tuning; 13.1 Automatic Gathering of Statistics; 13.2 The AWR and the ADDM; 13.3 Automating SQL Tuning
Part III: Physical and Configuration Tuning
Chapter 14. Tuning Oracle Database File Structures; 14.1 Oracle Database Architecture and the Physical Layer; 14.2 Tuning and the Logical Layer; 14.3 Automating Database File Structures; Chapter 15. Object Tuning; 15.1 Tables; 15.2 Indexes; 15.3 Index-Organized Tables and Clusters; 15.4 Sequences; 15.5 Synonyms and Views; 15.6 The Recycle Bin; Chapter 16. Low-Level Physical Tuning; 16.1 What Is the High-Water Mark?; 16.2 Space Used in a Database; 16.3 What Are Row Chaining and Row Migration?; 16.4 Different Types of Objects
16.5 How Much Block and Extent Tuning?

Sommario/riassunto

Tuning of SQL code is generally cheaper than changing the data model. Physical and configuration tuning involves a search for bottlenecks that often points to SQL code or data model issues. Building an appropriate data model and writing properly performing SQL code can give 100%+ performance improvement. Physical and configuration tuning often gives at most a 25% performance increase. Gavin Powell shows that the central theme of Oracle 10gR2 Performance Tuning is four-fold: denormalize data models to fit applications; tune SQL code according to both the data model and the application
